



DEVELOPMENT OF A FRONTEND FOR AN INFRASTRUCTURE AS A SERVICE CALLED OPENHUACA

A Degree Thesis

**Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona**

Universitat Politècnica de Catalunya

by

Carla Brugulat Rica

**In partial fulfilment
of the requirements for the degree in
TELECOMMUNICATIONS SYSTEMS ENGINEERING**

Advisor: Jose Luis Muñoz Tapia

Barcelona, May 2018

Abstract

Nowadays, web pages are part of the essence of the information society, there are many websites that offer the possibility to create pages in a simple way without having to know anything about programming or web development, but if what we want is to program our own web from scratch what technologies exist to be able to do it in the simplest and fastest way.

This project focuses on the study of several web technologies starting by the basic ones: HTML (1) *, CSS (2), JS (3), etc. followed by others that are more advanced level such as jQuery, React, Angular and finally the study of a content management site: Oracle WebCenter Sites.

After this study, everything learned is applied to the creation of the official OpenHuaca website, that is a cloud platform created by several final thesis students able to create containers, domains, ..., generating two different versions, the first with basic technologies and the second with React.

As a result, a comparison of the three variations of the web is made both at the level of knowledge required and at the functional level.

Resum

Actualment, les pàgines web formen part de l'essència de la societat de la informació, hi ha moltes webs que t'ofereixen la possibilitat de crear pàgines de forma senzilla sense necessitat de saber res sobre programació o desenvolupament web, però si el que volem és programar la nostra pròpia web des de zero quines tecnologies existeixen per poder fer-ho de la manera més senzilla i ràpida.

Aquest projecte es focalitza en l'estudi de diverses tecnologies web partint de les bàsiques: HTML (1) *, CSS (2), JS (3), etc. seguit d'altres que són de nivell més avançat com jQuery, React, Angular i finalment l'estudi d'un gestor de continguts web: Oracle WebCenter Sites.

Després d'aquest estudi, s'aplica tot el que s'ha après en la creació de la pàgina web Oficial d'OpenHuaca, plataforma "cloud" creada per diversos estudiants de tesi final de carrera, generant dues versions diferents, la primera amb tecnologies bàsiques y la segona amb React.

Com a resultat, es realitza una comparació de les tres variacions de la web tant a nivell de grau de coneixement necessari com a nivell de funcionalitats.

Resumen

Actualmente las páginas web forman parte de la esencia de la sociedad de la información, existen muchas webs que te ofrecen la posibilidad de crear páginas de forma sencilla sin necesidad de saber nada sobre programación o desarrollo web, pero y si lo que queremos es programar nuestra propia web des de cero que tecnologías existen para poder hacerlo de la forma más sencilla y rápida.

Este proyecto se focaliza en el estudio de varias tecnologías web partiendo de las básicas: HTML^{(1)*}, CSS⁽²⁾, JS⁽³⁾, etc. seguido de otras que son de nivel más avanzado cómo jQuery, React, Angular y finalmente el estudio de un gesto de contenidos web: Oracle WebCenter Sites.

Después de este estudio, se aplica todo lo aprendido a la creación de la página web Oficial de OpenHuaca, software creado por varios estudiantes de tesis final de carrera, generando dos versiones diferentes, la primera con tecnologías básicas y la segunda con React.

Finalmente, se realiza una comparación de las tres variaciones de la web tanto a nivel de grado de conocimiento necesario como a nivel de funcionalidades.

*todos los acrónimos están debidamente listados en el apartado de Glosario

Dedicatoria

Esta tesis se la quiero dedicar a las siguientes personas:

A mis padres, Paqui Rica y Jordi Brugulat, porque siempre están con la mano tendida para ayudar, apoyar y respetar mis locuras.

Rafael Genes Duran, por ser mi apoyo incondicional a lo largo de la carrera y un puntal fundamental en mi vida.

A mis abuelos, Pepi, Eudald y Carme, por ser mis segundos padres siempre apunto para hacerme feliz.

A mis suegros, Adolfo y María Jesús, por ser también un gran apoyo des del primer momento.

Y finalmente, a Jose Luis Muñoz Tapia, mi tutor, por acompañarme y guiarme a lo largo de este proyecto.

Agradecimientos

Me gustaría agradecer la colaboración, apoyo y ayuda que recibida por parte de dos personas en particular y por el equipo que conforma OpenHuaca.

- Jose Luis Muñoz y Rafa Genes Duran que me han guiado, me han dado sus mejores consejos y me han abierto las puertas a todo su conocimiento para poder seguir adelante.
- A todo el equipo OpenHuaca por prestarme sus conocimientos sobre el proyecto y estar siempre disponibles para responder preguntar y ayudar en lo que fuera necesario.

Historial de revisiones y aprobaciones

Revisión	Date	Propósito
0	22/04/2018	Creación del documento
1	29/04/2018	Primera edición del documento
2	6/05/2018	Segunda edición del documento
3	9/05/2018	Revisión final del documento

LISTA DE DISTRIBUCIÓN DE DOCUMENTOS

Nombre	e-mail
Carla Brugulat Rica	cbrugulat@gmail.com
Jose Luis Muñoz Tapia	jose.muñoz@entel.upc.edu

Escrito por:		Revisado y aprobado por:	
Fecha	06/05/2018	Fecha	09/05/2018
Nombre	Carla Brugulat Rica	Nombre	Jose Luis Muñoz Tapia
Puesto	Autor del proyecto	Puesto	Supervisor del proyecto

Tabla de contenido

Abstract	1
Resum	2
Resumen	3
Dedicatoria	4
Agradecimientos	5
Historial de revisiones y aprobaciones	6
Tabla de contenido	7
Listado de Figuras	9
Listado de Tablas	10
1. Introducción	11
1.1. Propósito del proyecto	11
1.2. Requerimientos y Especificaciones	11
1.3. Metodología y procedimientos	12
1.3.1 Comunicación	12
1.3.1. Documentación	12
1.3.2. Desarrollo	12
1.4. Plan de Trabajo	13
1.4.1. Paquetes de trabajo	13
1.4.2. Metas	15
1.4.3. Diagrama de Gantt	15
1.5. Incidencias y modificaciones	17
2. Estado del arte	18
3. Metodología y desarrollo del proyecto	19
3.1.1. Búsqueda de información	20
3.1.2. Aprendizaje y formación	22
3.1.3. Comparación de los lenguajes	23
3.1.4. Creación de la página web de OpenHuaca	23
3.1.5. Aspecto de la web oficial de OpenHuaca	24
3.1.6. Desarrollo con HTML, CSS y JS	26
El desarrollo de la web con HTML sigue la siguiente estructura:	26

3.1.7. Desarrollo con React	28
4. Resultados	34
5. Costes	36
6. Conclusiones y futuros desarrollos	37
Bibliografia.....	38
Glosario	39

Listado de Figuras

Este apartado contiene la lista completa de Figuras que aparecen en los diferentes apartados que conforman esta tesis.

- Fig 1. Diagrama de Gantt Febrero
- Fig 2. Diagrama de Gantt Marzo
- Fig 3. Diagrama de Gantt Abril
- Fig 4. Diagrama de Gantt Mayo
- Fig 5. Ejemplo funcionamiento HTML y CSS
- Fig 6. Ejemplo funcionamiento jQuery
- Fig 7. Home OpenHuaca
- Fig 8. AboutUs OpenHuaca
- Fig 9. News OpenHuaca
- Fig 10. Functionalities OpenHuaca
- Fig 11. Estructura básica HTML y aspecto del <head> de la página web de OpenHuaca
- Fig 12. Ejemplo definición del CSS con algunos conceptos básicos
- Fig 13. Ejemplo carpeta de proyecto React
- Fig 14. Contenido package.json con todas las dependencias y configuración servidor
- Fig 15. Configuración final Babel
- Fig 16. Configuración final Webpack
- Fig 17. Ejemplo clase React
- Fig 18. Ejemplo de componente que se llama en la clase de la Figura 17

Listado de Tablas

Este apartado contiene la lista completa de Tablas que aparecen en los diferentes apartados que conforman esta tesis.

- Tabla 1. Paquete de Trabajo: Documentación
- Tabla 2. Paquete de Trabajo: Supervisión
- Tabla 3. Paquete de Trabajo: Desarrollo
- Tabla 4. Metas
- Tabla 5. Dependencias básicas para React
- Tabla 6. Dependencias especiales para React-Webpack

1. Introducción

En este apartado se recogen los objetivos principales, las especificaciones, la metodología y plan de trabajo, así como las incidencias surgidas durante el desarrollo del proyecto.

1.1. Propósito del proyecto

El propósito principal de este proyecto es establecer una comparación de diferentes tecnologías web que existen actualmente en el mercado, partiendo de la base del desconocimiento de dichas y con la intención de crear una web de la forma más fácil y sencilla a la vez que más funcional y dinámica.

Las técnicas que se van a estudiar se pueden dividir en tres grupos:

- Tecnologías básicas
- Tecnologías basadas en JavaScript
- Gestores de contenidos

Este proyecto final de carrera en su mayor parte tiene un carácter teórico que permite acabar recreando la página web oficial del software OpenHuaca.

Para poner a prueba los conocimientos obtenidos en la primera parte del proyecto se realizan tres webs que tienen el mismo aspecto pero que están basadas en tecnologías diferentes, poniendo en práctica los resultados de la comparación.

1.2. Requerimientos y Especificaciones

Para realizar este proyecto se establecieron dos requisitos básicos, partir de la base del desconocimiento en tecnologías web para así tener una visión completamente objetiva a la hora de buscar y escoger las tecnologías que se iban a trabajar y en segundo lugar, que esas tecnologías permitieran crear una web funcional tanto para incluir funcionalidades básicas como otras de más avanzadas.

El conjunto de elementos que debe tener la web Oficial de OpenHuaca conforman las especificaciones de este proyecto y se listan a continuación:

- Sección de descarga para que se pueda descargar el software OpenHuaca, junto con los datos de contacto del proyecto.

- Sección de noticias con un blog que incluye las últimas novedades del proyecto global de OpenHuaca.
- Sección explicativa sobre qué es OpenHuaca y quién forma parte del proyecto.
- Sección de apoyo para todo aquel que quiera usar la plataforma disponga de una guía de uso.

1.3. Metodología y procedimientos

Este apartado recoge el conjunto de métodos utilizados para llevar a cabo el proyecto y así garantizar su desarrollo y finalización satisfactoriamente. Los procedimientos seguidos se pueden dividir en tres subgrupos:

1.3.1 Comunicación

Conjunto de reuniones con el tutor, una vez por semana, y con el equipo que conforma OpenHuaca, cada quince días, a lo largo de la realización del proyecto.

1.3.1. Documentación

A lo largo del proyecto, y para ir preparando la tesis (este documento) se resumían los resultados semanales en un documento compartido con el tutor.

1.3.2. Desarrollo

Para realizar la parte práctica de este proyecto se utilizó el *Visual Studio Code* como editor de código y se fue subiendo todo el proyecto en una máquina virtual de la plataforma OpenHuaca para que todos los miembros del equipo pudieran ir visualizando los avances.

1.4. Plan de Trabajo

El proyecto se ha dividido en tres paquetes de trabajo para que su estructura temporal fuera más sencilla. A continuación, se describen estos subgrupos de trabajo y se listan las metas propuestas desde el primer momento junto con un Diagrama de Gantt que permite visualizar la planificación de forma clara y concisa.

1.4.1. Paquetes de trabajo

Project: DEVELOPMENT OF A FRONTEND FOR AN INFRASTRUCTURE AS A SERVICE CALLED OPENHUACA	PT ref: #1 Documentación
Componente principal: Documentación	
Descripción corta: Redacción de los resúmenes semanales y de todos los documentos relacionados con la realización del proyecto.	Fecha inicio: 05/02/2018 Fecha final: 23/05/2018
Entregas	Fechas
Resúmenes semanales	Cada semana
Tesis Final	11/05/2018
PowerPoint Presentación	21/05/2018

Tabla 1. Paquete de Trabajo: Documentación

Project: DEVELOPMENT OF A FRONTEND FOR AN INFRASTRUCTURE AS A SERVICE CALLED OPENHUACA	PT ref: #2 Supervisión
Componente principal: Supervisar el estado proyecto	
Descripción corta: Reuniones semanales tanto con el tutor cómo con el equipo OpenHuaca, además de comprobar y actualizar la planificación del proyecto.	Fecha inicio: 05/02/2018 Fecha final: 23/05/2018
Entregas	Fechas
Actualización plan de trabajo 1	13/03/2018
Actualización plan de trabajo 2	17/04/2018
Actualización plan de trabajo 3	04/05/2018

Tabla 2. Paquete de Trabajo: Supervisión

Project: DEVELOPMENT OF A FRONTEND FOR AN INFRASTRUCTURE AS A SERVICE CALLED OPENHUACA	PT ref: #3 Desarrollo
Componente principal: Desarrollar el proyecto	
Descripción corta: Write documentation files	Fecha inicio: 05/02/2018 Fecha final: 23/05/2018
Entregas	Fechas
OpenHuaca WebSite con tecnologías básicas	Inicio Marzo
OpenHuaca WebSite con React	Inicio Abril
OpenHuaca WebSite con OWCS	Inicio Mayo

Tabla 3. Paquete de Trabajo: Desarrollo

1.4.2. Metas

PT#	Título	Entregas	Fecha
1	Idea de proyecto	Plan de trabajo	6/02/2018
1	Resumen comparativas 1	Comparación tecnologías básicas	Final Febrero
1	Resumen comparativas 2	Comparación tecnologías JavaScript	Final Marzo
1	Resumen comparativas 3	Comparación gestores de contenidos	Final Abril
1	Tesis Final de Carrera	Tesis (este documento)	11/05/2018
1	Presentación Final	PowerPoint de la presentación final	21/05/2018
2	Supervisión plan de trabajo 1	Primera actualización plan de trabajo inicial	13/03/2018
2	Supervisión plan de trabajo 2	Segunda actualización plan de trabajo inicial	17/04/2018
2	Supervisión plan de trabajo 3	Tercer actualización plan de trabajo inicial	04/05/2018
3	OpenHuaca Official WebSite 1	Web con tecnologías básicas	Inicio Marzo
3	OpenHuaca Official WebSite 2	Web con React	Inicio Abril
	OpenHuaca Official WebSite 3	Web con Oracle WebCenter Sites	Inicio Mayo

Tabla 4. Paquete de Trabajo: Metas

1.4.3. Diagrama de Gantt

Las siguientes imágenes muestran la división de tareas por meses de cada paquete de trabajo:

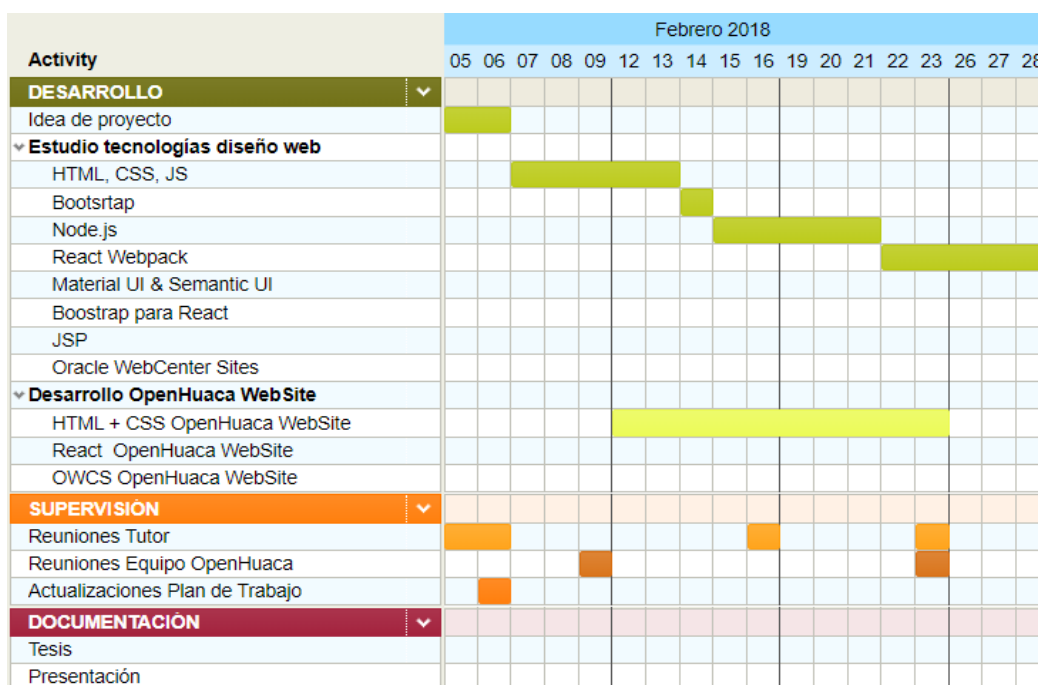


Fig 1. Diagrama de Gantt Febrero

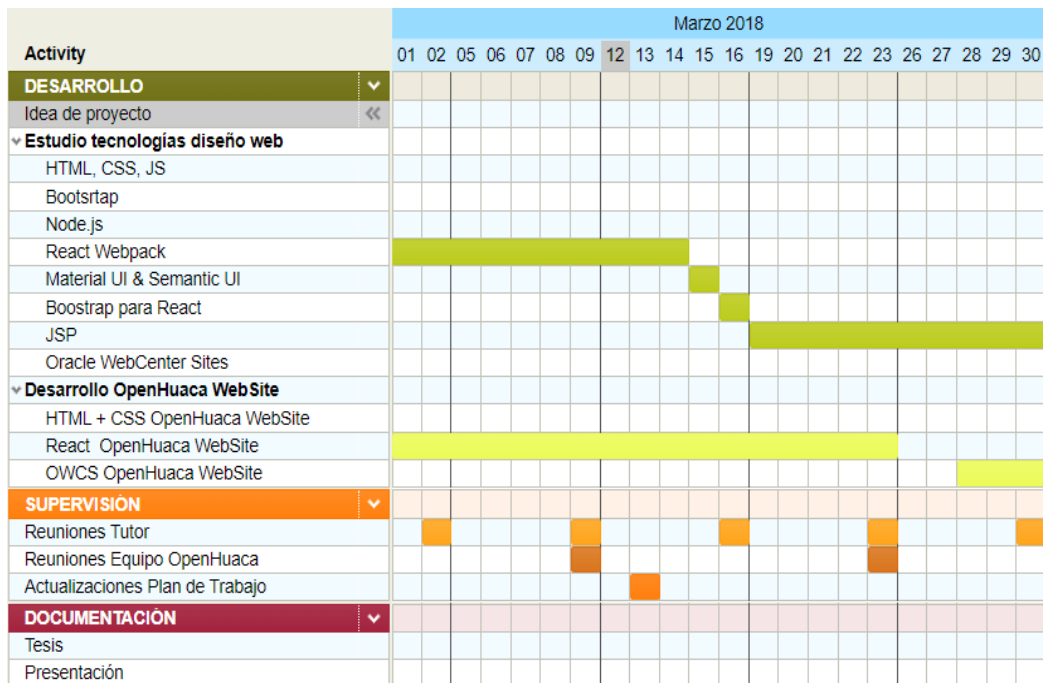


Fig 2. Diagrama de Gantt Marzo

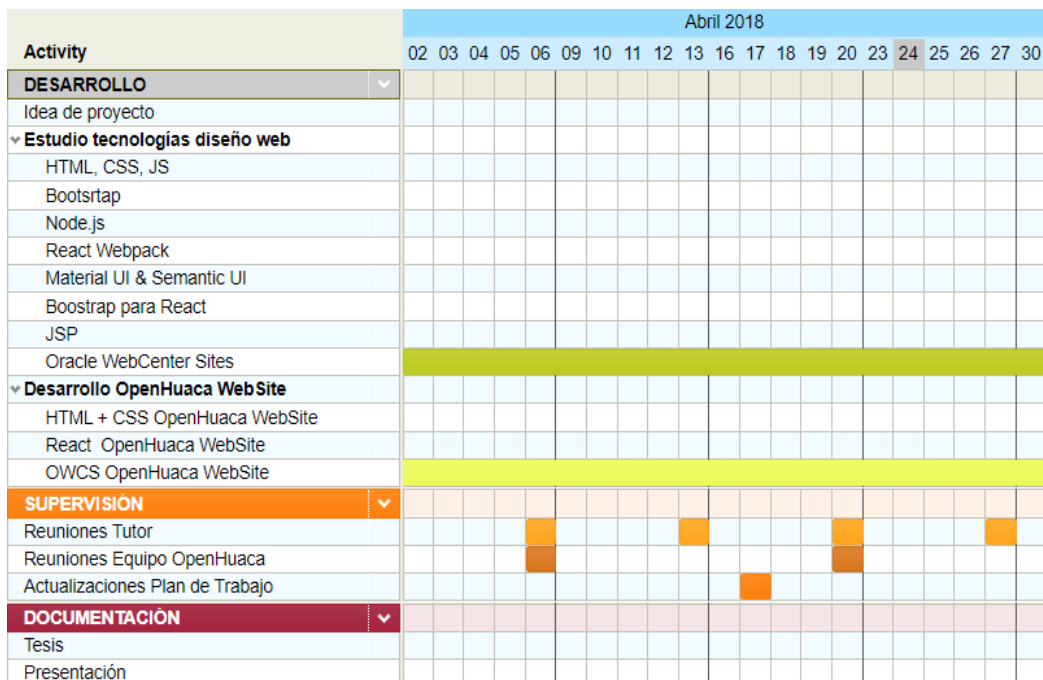


Fig 3. Diagrama de Gantt Abril

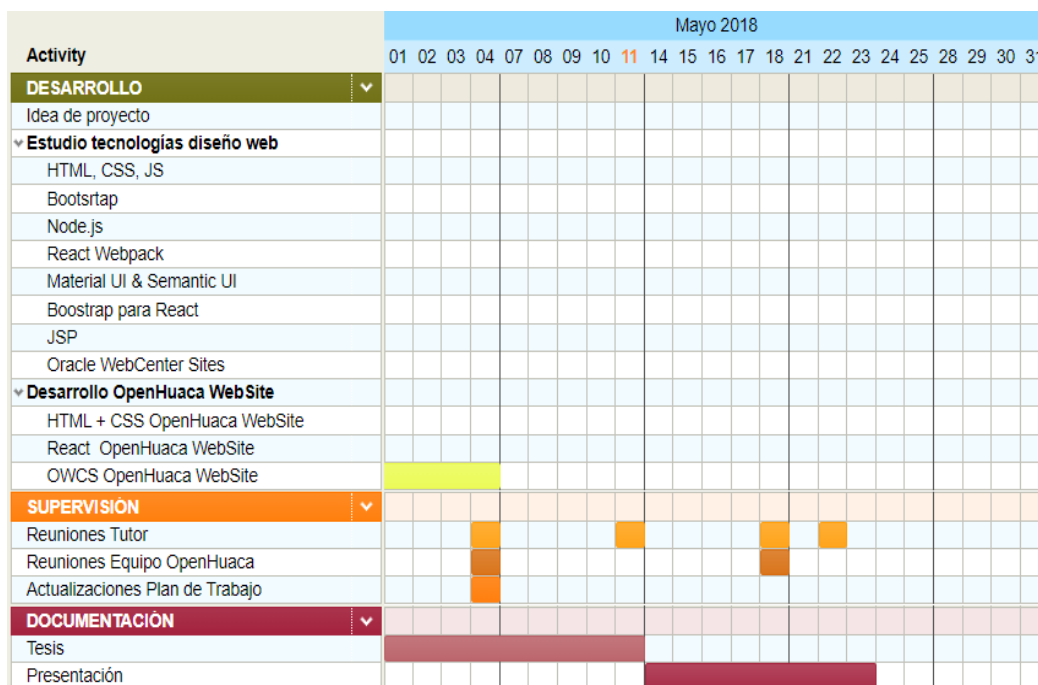


Fig 4. Diagrama de Gantt Mayo

1.5. Incidencias y modificaciones

Este apartado lo voy a dividir en dos partes, la primera para justificar las semanas dedicadas a la realización de este proyecto y la segunda para hablar del cambio de planteamiento de este proyecto respecto al inicial.

Esta tesis se ha llevado a cabo a lo largo de 15 semanas aproximadamente, del 5 Febrero de 2018 al 23 de Mayo de 2018, ya que con anterioridad cursaba un TFG completamente diferente al de esta tesis, el cual decidí abandonar por motivos personales.

Cuando se me planteó la idea de este TFG, inicialmente consistía en crear la web oficial de OpenHuaca con las tecnologías que considerara más oportunas, pero después de un par de semanas de realización con el tutor del proyecto nos dimos cuenta de la importancia y de la cantidad de tecnologías existentes que me ofrecían distintas características a la hora de crear la página web; por ello cambiamos el enfoque inicial para uno de más teórico y docente donde la parte web fuera sólo la puesta en práctica de los resultados obtenidos.

Otro cambio producido a lo largo de la realización del proyecto es que inicialmente se pretendía recrear un site con OWCS pero finalmente, con el tutor se desestimó esta idea ya que el formato era de pago.

2. Estado del arte

En nuestra sociedad actual, las páginas web forman parte de nuestro mundo cotidiano al igual que muchas otras tecnologías de la información, pero seguramente son la más extendida de las que están directamente unidas a la red.

Las tecnologías que nos permiten realizar “websites” son muchas y variadas tanto a conceptualmente cómo a en desarrollo, algunas más fáciles de aprender y otras de nivel más avanzado, pero para este proyecto nos interesa evaluar esas que sin saber nada sobre programación web parecen más fáciles de utilizar, por eso en este apartado nos centramos en los lenguajes de programación más básicos que existen actualmente.

Por supuesto debemos partir de la base que conforma cualquier página web: HTML que es el elemento de construcción más básico de una página web y se usa para crearla y representarla visualmente. Determina el contenido de la página web, pero no su funcionalidad. Este lenguaje tampoco nos permite estilizar ni decorar nuestra web, para ello principalmente se usa CSS.

Pero ninguno de estos dos lenguajes nos permite añadir funcionamiento a nuestra web, por ejemplo: el HTML nos permitiría crear un botón y el código CSS nos permitiría pintarlo de color azul pero ninguno de estos dos indicaría que hacer en el caso de que este botón se pulse.

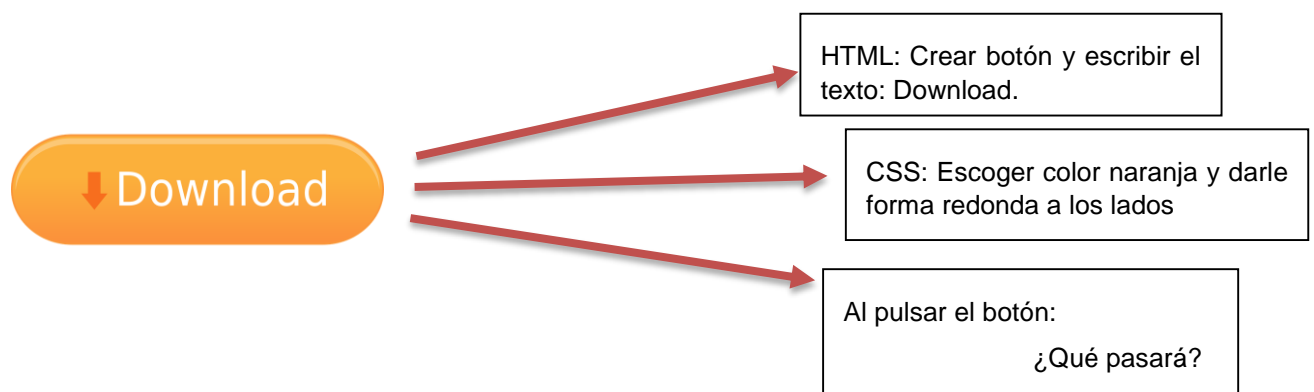


Fig 5. Ejemplo funcionamiento HTML y CSS

Para añadir funcionalidad a este botón y a cualquier elemento de nuestra página web se usa JavaScript que es un lenguaje de programación orientado a objetos y es un lenguaje muy ligero conocido popularmente como el lenguaje de scripts para páginas web.

Gracias a JS, podremos indicarle al botón de nuestro ejemplo que cuando sea pulsado descargue el contenido deseado.

A partir de aquí los lenguajes de programación que existen son muchos: PHP, ASP, Angular, JQuery, Python, React, JSP, etc.

De todos ellos, dejando aparte los que comentaremos en el siguiente apartado quiero destacar PHP, lenguaje de programación para el desarrollo web con contenido dinámico, es decir que añade funcionalidad a nuestra web, lo mismo que define JS. Este fue uno de los primeros lenguajes de programación y su característica más destacada es que ejecuta su código al lado del servidor. Actualmente, y aunque es uno de los lenguajes más usados en Internet, está cayendo en desuso ya que la mayoría de aplicaciones nuevas se están empezando a diseñar con “frameworks” de JavaScript, de los cuales hablaremos a continuación.

3. Metodología y desarrollo del proyecto

Para elaborar este TFG se partía de la base de no saber nada sobre programación y desarrollo web, así que el primer paso era informarse y buscar información sobre las posibles tecnologías que se podían utilizar. Como se ha comentado con anterioridad, la lista de lenguajes de programación es bastante grande pero muchos de ellos parten de otros como base.

Esta sección se va a dividir en cuatro partes:

- Búsqueda de información.
- Aprendizaje y formación.
- Comparación de los lenguajes.
- Creación de la página web de OpenHuaca.

Pero antes la descripción de la metodología general que se ha seguido a lo largo del proyecto, tanto a nivel de comunicación con el tutor, como a nivel seguimiento de la planificación.

El proyecto se empezó a desarrollar en Febrero, cuando el equipo que conforma OpenHuaca decide que necesitan una web para acercar la plataforma a los usuarios y me proponen el proyecto ya que el manager del equipo sabía mi interés por el mundo del desarrollo web. El proyecto se me plantea como un reto ya que no tenía formación ninguna en tecnologías web, pero aun así acepto y empieza la búsqueda de información, a las pocas semanas decido que este proyecto se convierta finalmente en mi tesis final de carrera.

3.1.1. Búsqueda de información

En esta primera etapa, después de haber realizado una búsqueda inicial, se marcan las metas a seguir y la planificación en semanas del proyecto.

Una vez aceptado el proyecto, el primer paso fue obtener toda la información posible sobre tecnologías web y desarrollo de páginas dinámicas y completamente funcionales. A la hora de buscar es muy importante saber reconocer que páginas son verídicas y que otras carecen de sentido para lo que necesitas realizar. Las bases para realizar la web quedan claras: HTML, CSS y JS. Pero a partir de aquí hay muchos caminos a escoger, una página web se puede crear simplemente con estos tres lenguajes de programación, pero ¿me interesa para mi proyecto en particular?

Durante la búsqueda encuentro muchas opciones que me permitirían añadir otras funcionalidades extra a mi proyecto si escojo un lenguaje de programación diferente o derivado de uno de los básicos. Entre las más destacadas y usadas por los usuarios se encuentra jQuery, React y Angular, a continuación se explican sus principales similitudes y diferencias.

La primera cosa en común es que las tres están basadas en JS, es decir son frameworks de JavaScript, y por lo tanto permiten añadir funcionalidad a cualquier página web, pero cada una de ellas lo hace de forma bastante diferente.

Empezando por JQuery, esta permite la visualización de los diferentes elementos creados en el HTML des de su propio código, concepto de gran utilidad ya que nos permite renderizar una página HTML por partes, véase ejemplo a continuación.

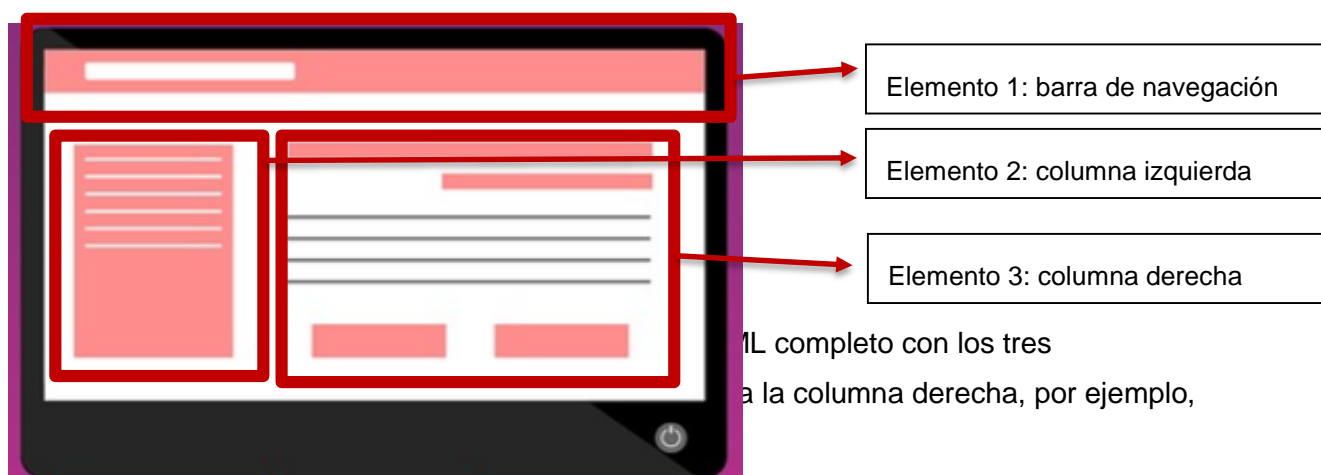


Fig 6. Ejemplo funcionamiento jQuery

Pero actualmente, existe la versión de CSS³ que te permite realizar las mismas funcionalidades, por esa razón su utilización ya no es tan extendida cómo antes, no obstante es una forma fácil y rápida de crear un site dinámico.

El segundo framework destacado es Angular que fue el más utilizado del mercado hasta el momento, sus dos características más destacadas son que permite intercambiar información de HTML a JavaScript y viceversa de una forma muy sencilla, haciendo posibles las aplicaciones de una sola página y que se basa en el Modelo-Vista-Controlador, es decir por un lado define componentes para la representación de la información, y por otro lado para la interacción con el usuario.

Inicialmente Angular solo permitía crear webs de una sola página pero actualmente sus últimas versiones ya permiten diseños de páginas múltiples.

Por último React, este es el framework más nuevo de los tres, a diferencia de Angular utiliza Modelo-Vista, es decir tenemos un modelo y una vista que interactúan entre ellos pero no sabemos cómo lo hacen. Uno de los puntos fuertes de React es que permite generar componentes reutilizables para otros proyectos. A partir de aquí la idea de desarrollo web vista hasta el momento da un cambio: en lugar de escribir páginas que formaran parte de un site, escribimos componentes que manejaran otros componentes. La parte más negativa de React es que en un primer momento, para un programador que se inicia no es algo tan fácil de dominar cómo los casos anteriores.

Dentro del mundo del desarrollo de páginas web, a parte de los lenguajes de programación existen un seguido de herramientas de ayuda de código libre para programadores, entre ellas quiero destacar tres:

- [Bootstrap](#) que es un conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales.
- [Google Fonts](#) que es una librería con más de 800 fuentes (tipologías de letras) que se pueden incluir de forma sencilla a cualquier web a través de CSS.
- [Colours NeilOrangePeel](#) que es un diccionario con el recopilatorio de colores utilizable en CSS por nombre

Finalmente, dentro de la búsqueda de información aparece un nuevo concepto llamado Gestor de Contenidos, es una herramienta/programa informático que permite crear una estructura de soporte (framework) para la creación y administración de contenidos web, por parte de los programadores pero también por parte de los administradores, editores, participantes y demás usuarios. Este concepto abre una nueva puerta en el desarrollo web hacia convertirlo en un proceso colaborativo entre diferentes partes. Por ejemplo, si nos planteamos que somos una redactora de un diario digital y usamos un gestor de contenidos, las personas que redactan los artículos podrán ser las mismas que suban ese contenido a la web sin saber nada de programación ya que el gestor actuará de interface.

Dentro del conjunto de gestores de contenidos, las opciones opensource son más reducidas. Así que, dentro de las de pago encuentras opciones interesantes cómo la dos que ofrece la empresa Oracle:

- Oracle WebCenter Sites (OWCS)

Producto que permite gestionar la experiencia en la página web des de diferentes puestos: marketing, usuarios, administradores, ... todo ello de forma colaborativa y a través de una interfaz intuitiva.

3.1.2. Aprendizaje y formación

Para poder responder a esa primera pregunta, empiezo mi formación en desarrollo web con un curso online llamado “Web Developer BootCamp” del portal de cursos on-line Udemy, que contiene el conjunto de técnicas necesarias para poder desarrollar una web de principio a fin sin muchas complicaciones y con una gran variedad de funcionalidades.

Con este curso llego a la conclusión de poder realizar la página web de OpenHuaca con sus diferentes especificaciones simplemente con la utilización de tecnologías básicas, pero más adelante, fuera del rango del proyecto, esta web deberá añadir más funcionalidades cómo pueden ser un formulario de contacto o una autenticación de usuario. Para ello necesitaría una tecnología que me permitiera comunicación con el backend, por esa razón me decanto por React, ya que es una tecnología basada en Node.js que nos permite la comunicación con el backend.

En el mismo portal de cursos, Udemy, realizo un curso basado en React para desarrollar el portal de OpenHuaca.

Para la parte de gestión de contenidos, para obtener más información sobre Oracle WebCenter Sites existe un tutorial gratuito en el propio portal de Oracle que te permite descargar una versión local de la interfaz y realizar un site paso a paso.

3.1.3. Comparación de los lenguajes

Una vez realizada toda la búsqueda y haber acabado con la formación, puedo realizar una comparación de los diferentes lenguajes vistos y de qué forma estos nos pueden resultar más útiles dependiendo de las especificaciones que necesite nuestro “website”.

En este apartado resumo esta comparación con los puntos más importantes y en el apartado de Resultados se incluye una tabla comparativa más extensa.

A nivel de dificultad a la hora de crear una página web dinámica, la opción más fácil sería usar HTML, CSS junto con JQuery. Pero si miramos a nivel de posibilidades en creación de contenidos en la web la mejor opción sería React ya que nos permite esa conexión con el backend de forma más cómoda.

En segundo lugar, si lo que queremos es hacer una aplicación de una sola página el más aconsejable sería Angular ya que nos permite hacerlo de forma mucho más clara y sencilla que React, que cómo ya se ha comentado puede resultar un lenguaje de programación pesado a nivel inicial.

3.1.4. Creación de la página web de OpenHuaca

Para poner en práctica todo lo aprendido en los apartados anteriores, se realizan tres páginas web que van a tener el mismo aspecto pero que cada una va a estar realizada con un lenguaje de programación diferente.

Este apartado se divide en 3 subsecciones:

- Aspecto de la web oficial de OpenHuaca.
- Desarrollo con HTML, CSS y JS.
- Desarrollo con React.

Pero primero, unas aclaraciones inicial, las webs utilizan Bootstrap cómo ayuda a la programación de algunas partes del site, en cada apartado se cuenta cómo se ha tenido que incluir para que funcione correctamente con cada lenguaje de programación. Y en segundo lugar, en cada sección al final se hace un recopilatorio de los problemas encontrados al programar la web.

3.1.5. Aspecto de la web oficial de OpenHuaca

El aspecto de la web se puede ver en las siguientes capturas, es importante recordar que las especificaciones y el contenido que debe tener la web es el que se describe en el apartado de Introducción/Requisitos y especificaciones.

Primeramente dispone de una página de inicio que contiene un carrusel, diseño de Bootstrap, con links a diferentes secciones de la web. Seguido de un link de descarga del software y de toda la información de contacto de OpenHuaca.



Fig 7. Home OpenHuaca

La segunda página de la web contiene toda la información relacionada con el proyecto OpenHuaca: que es OpenHuaca, una galería de imágenes con los miembros del equipo y el porqué de su desarrollo.



Fig 8. AboutUs OpenHuaca

La tercera sección incluye un blog con noticias de interés para aquellos que quieran formar parte del proyecto OpenHuaca o quieran descargar y utilizar el código.

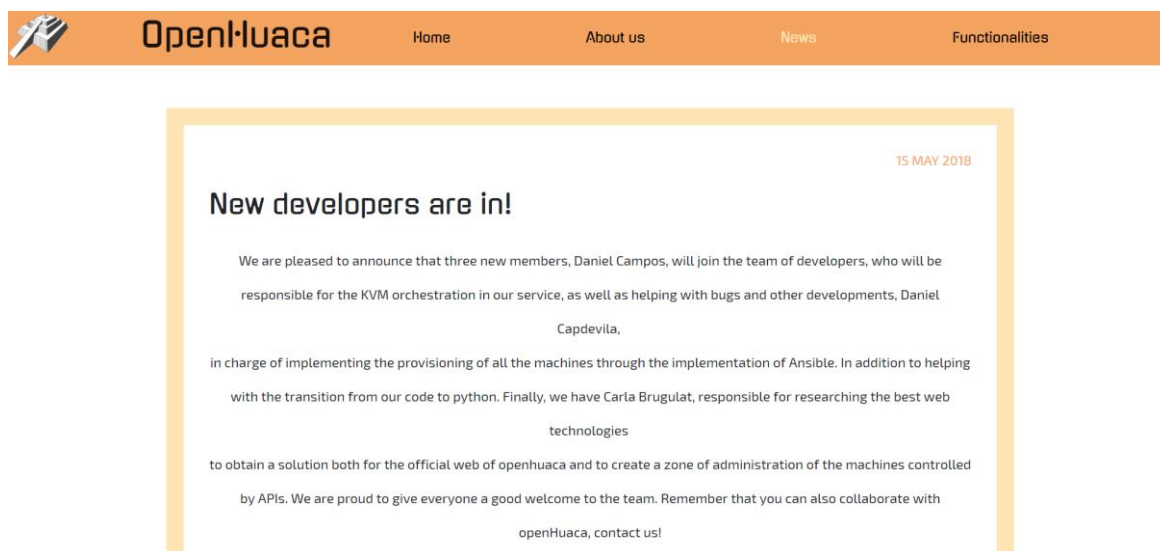


Fig 9. News OpenHuaca

Finalmente, el último apartado de la barra de navegación nos lleva a la sección de funcionalidades, dónde el usuario podrá encontrar una guía práctica de la plataforma OpenHuaca; por ejemplo, podrá encontrar información sobre cómo crear o destruir dominios, cómo generar certificados, etc.

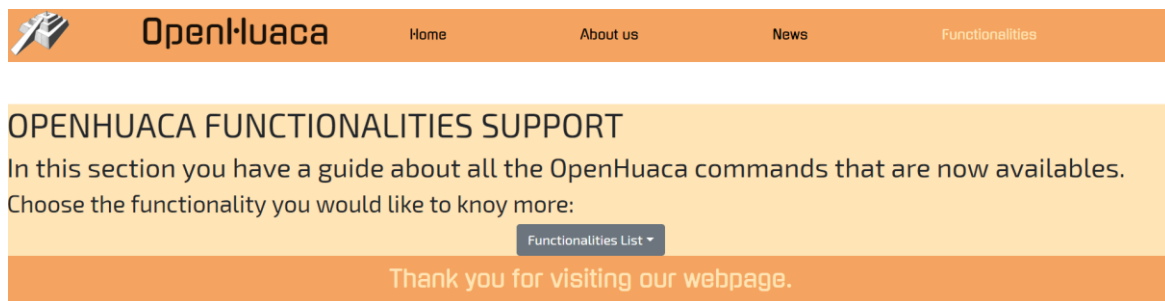


Fig 10. Functionalities OpenHuaca

3.1.6. Desarrollo con HTML, CSS y JS

El desarrollo de la web con HTML sigue la siguiente estructura:

Cada página se renderiza en un HTML distinto y deberá incluir todos los componentes que esa página necesita, este concepto no es muy práctico ya que tendremos un elemento que se visualiza en todas las páginas, cómo puede ser la barra de navegación, definido muchas veces.

La estructura de cada página viene marcada por la estructura del lenguaje HTML, cómo se puede visualizar en la imagen:



Fig 11. Estructura básica HTML y aspecto del <head> de la página web de OpenHuaca

Cómo se observa en el “head” de la imagen superior se requieren los diferentes estilos (CSS), los elementos que agregan funcionalidad al site (JS) y otras informaciones esenciales para el navegador.

Para el caso del site OpenHuaca se requiere incluir: el CSS con los estilos creados específicamente para esta él, los archivos requeridos por Bootstrap y las referencias a Google Fonts. Para el correcto funcionamiento de las herramientas de Bootstrap, en este desarrollo, se deben incluir las funcionalidades y el CSS que se muestran en la imagen superior.

La estructura del CSS se puede apreciar en la siguiente captura de una parte del código de Styles.css creado específicamente para esta web.

```

/* OpenHuaca Official Website Styles*/

/* ..... GENERAL .....*/
body {
  font-family: 'Gugi', cursive;
  text-align: right;
}

.Home, .Functionalities {
  background:moccasin;
}

.Information, .News,{
  background:none;
}

p {
  text-align: left;
  font-size: 15px;
  margin-top: 15px
}

a {
  font-family: 'Gugi', cursive;
  margin-left: 80px;
  margin-right: 80px;
  text-decoration: none;
  color:brown;
}

h1, h2, h3, h4, h5, h6{
  text-align: left;
}

```

Estilos referidos al contenido que hay dentro del tag de la estructura de HTML:

 <body> </body>

Estilos referidos a las diferentes páginas del site referenciadas por classe:

 <div class="Home"> </div>

Estilos referidos a todos los tags de tipo <p> o <a> o <h..> .

Fig 12. Ejemplo definición del CSS con algunos conceptos básicos

Durante el desarrollo de esta parte, las dificultades que me he encontrado son básicamente de tipo disponibilidad de funciones y dinamismos, cómo por ejemplo el hecho de no poder separar el pie de página en un documento HTML a parte e incluirlo en el documento de la página con un simple link o algo parecido. Por otro lado, aprender HTML, CSS y JS no es una tarea difícil y esto permite que se pueda realizar todo con mucha rapidez.

3.1.7. Desarrollo con React

La realización de la web utilizando React, fue una tarea más costosa y laboriosa que en el caso anterior, empezando por la instalación y el entorno que debes preparar antes de empezar a programar.

En el caso anterior, el inicio de programación es inmediato, el único paso a considerar es la inclusión de Bootstrap para que funcionen los elementos realizados con esta librería. Pero en el caso de React, antes de ponerse a programar hay que descargar, instalar y realizar un seguido de pasos para que la web vaya a funcionar correctamente.

Antes de empezar con los pasos, quería comentar que aparte de Bootstrap existen otras librerías con código específico para React cómo pueden ser Semantic.ui o Material.ui pero tomé la decisión de trabajar con Bootstrap para que las dos páginas utilizaran las mismas herramientas y así poder analizar tanto su configuración cómo su funcionamiento.

El primer paso es instalar Node.js ya que es la base de la programación de React, seguido de la instalación de dos componentes más: yarn o npm que son dos gestores de paquetes de tipo JavaScript, el segundo es el que va por defecto con Node.js, si se programa en Linux se recomienda usar npm ya que el primero puede dar errores de configuración.

Una vez instalado uno o otro, en mi caso npm, se puede empezar con la configuración de entorno y dependencias que necesitara React para poder entender y trabajar con la web que vamos a crear. Todas estas instalaciones se realizan a través del terminal en la ruta referente a la carpeta dónde se vaya a guardar el código del proyecto.

Las mínimas dependencias que se necesitan son las siguientes:

Dependencia	Uso	Comando de descarga
Babel-cli	Compilador especial para React, que nos permite escribir y desarrollar el código de forma más cómoda y con menos posibilidad de error.	npm install babel-cli
Babel-core		npm install babel-core
Babel-preset-react		npm install babel-preset-react
Babel-preset-env		npm install babel-preset-env
React	Permite el lenguaje react	npm install react
React-dom	Permite el lenguaje React con interacciones con el backend.	npm install react-dom

Tabla 5. Dependencias básicas para React

A parte de las anteriores, para este proyecto se han requerido dos instalaciones extras:

Dependencia	Uso	Comando de descarga
WebPack	Configuración web para sites con más de una página	npm install webpack
Webpack-dev-server	Servidor para subir la web a Internet especial para webpack	npm install webpack-dev-server
Webpack devtool	Permite analizar los errores de React directamente en la consola del navegador web.	npm install devtool
Bootstrap	Configuración de Bootstrap en React	npm install bootstrap jquery popper.js

Tabla 6. Dependencias especiales para React-Webpack

Una vez instaladas todas las dependencias, la carpeta del proyecto debe tener un aspecto similar al siguiente:

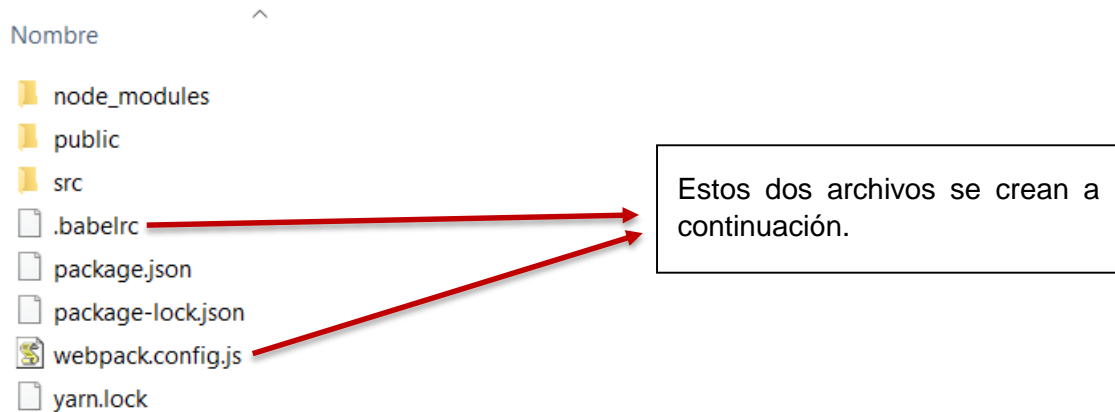
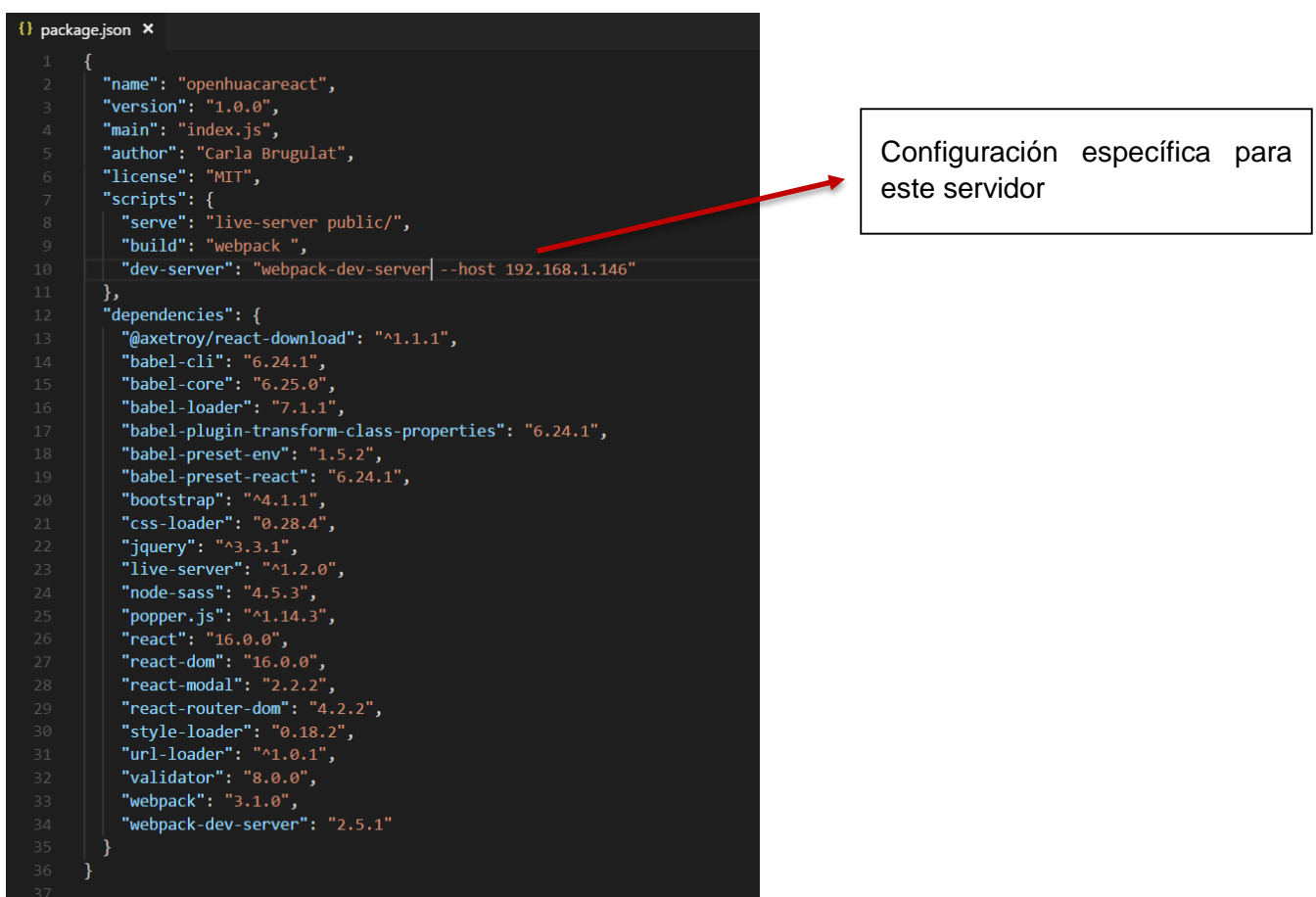
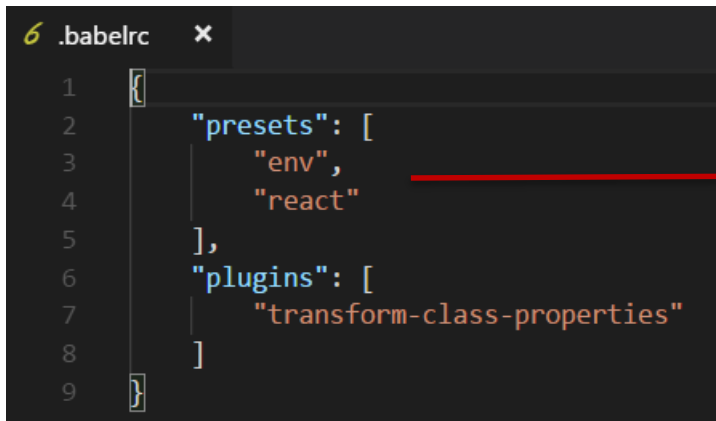


Fig 13. Ejemplo carpeta de proyecto React

A continuación se debe comprobar que dentro del archivo package.json aparece un listado con todas las dependencias y su número de versión, al mismo tiempo que se debe adecuar el archivo para que tengo un aspecto similar al de la siguiente captura:



Para acabar con la configuración de entorno, en la ruta dónde se han realizado todas las instalaciones deberemos crear dos archivos nuevos que contengan la información cómo en las imágenes de a continuación:



```

1  {
2    "presets": [
3      "env",
4      "react"
5    ],
6    "plugins": [
7      "transform-class-properties"
8    ]
9  }

```

Finalización de la configuración del compilador BABEL.

Fig 15. Configuración final Babel



```

1  const path = require('path');
2
3  module.exports = {
4    entry: './src/app.js',
5    output: {
6      path: path.join(__dirname, 'public'),
7      filename: 'bundle.js'
8    },
9    module: {
10     rules: [{
11       loader: 'babel-loader',
12       test: /\.js$/,
13       exclude: /node_modules/
14     }, {
15       test: /\.css$/,
16       use: [
17         'style-loader',
18         'css-loader',
19       ]
20     }, {
21       test: /\.?(png|svg|jpg|gif)$/,
22       use: [
23         'url-loader'
24       ]
25     }
26   ],
27   devtool: 'cheap-module-eval-source-map',
28   devServer: {
29     contentBase: path.join(__dirname, 'public'),
30     historyApiFallback: true
31   }
32 };

```

Fig 16. Configuración final Webpack

Cómo se puede observar, la preparación para poder empezar a programar hace que este estilo de lenguaje sea más incómodo al principio sobretodo.

Tal y cómo se ha comentado en apartados anteriores la mejor característica que tiene React es que nos permitirá crear componentes que luego añadiremos a nuestras páginas para acabar componiendo un componente mayor, de esta forma todo el código será futuramente reutilizable para otras webs o aplicaciones.

Por ejemplo, si recordamos la imagen del principio de jQuery (Fig. 5) y viendo el aspecto de la página de inicio (Fig.) podremos crear 5 componentes diferenciados:

- Barra de Navegación
- Carrusel
- Link de Descarga
- Información de Contacto
- Pie de página

De tal forma que cuando creemos las otras páginas del site podremos reaprovechar los componentes de la barra de navegación y del pie de página.

A continuación se muestran dos capturas de ejemplo sobre cómo se crea un componente y cómo este se llama en otro.

```

JS AboutUs.js x
1  import React from 'react';
2
3  import NavBar from './NavBar'
4  import What from './WhatisOH'
5  import Who from './WhoisOH'
6  import Why from './WhyisOH'
7  import Footer from './Footer'
8
9  export default class AboutUs extends React.Component{
10     render() {
11
12         return (
13             <div class="AboutUs">
14                 <div id="main">
15                     <div id="container">
16                         <What />
17                         <Who />
18                         <Why />
19                     </div>
20                     <div id="footer">
21                         <Footer />
22                     </div>
23                 </div>
24             </div>
25         )
26     }
27 }

```

Importación de todos los componentes que voy a usar para generar la clase AboutUs.

Exportación de la clase AboutUs para que se pueda usar en otras páginas como componente.

Contenido que se debe renderizar de la clase AboutUs cuando se invoque.

Estructura del código a renderizar muy similar al código HTML.

Fig 17. Ejemplo clase React

```
WhoisOH.js x
4 import Member2 from '../images/Team/Rafa.png'
5 import Member3 from '../images/Team/Jorge.png'
6 import Member4 from '../images/Team/Dani.png'
7 import Member5 from '../images/Team/Daniel.png'
8 import Member6 from '../images/Team/Carla.png'
9
10 export default class Carousel extends React.Component{
11   render() {
12     return (
13       /*<div className="jumbotron jumbotron-fluid" id="who">*/
14       <div className="who" id="jumbotron">
15         <div className="container">
16           <h2 className="display-4"> Who is behind OpenHuaca project? </h2>
17           <p className="lead">OpenHuaca is formed by the following group of professionals: </p>
18
19           <img className="galleryTeam" src={Member1}/>
20           <img className="galleryTeam" src={Member2}/>
21           <img className="galleryTeam" src={Member3}/>
22
23           <img className="galleryTeam" src={Member4}/>
24           <img className="galleryTeam" src={Member5}/>
25           <img className="galleryTeam" src={Member6}/>
26         </div>
27       </div>
28     )
29   }
30 }
31 ;;
```

Fig 18. Ejemplo de componente que se llama en la clase de la Figura 17

A la hora de integrar el css al site, React ofrece dos posibilidades, descargar un loader para poder usar un código css normal o descargar un loader para scss que es la tipología de css expresa para React. Para este proyecto, escogí la primera opción porque creo que es la más cómoda ya que así me permitía reutilizar el código css creado en la primera web.

Para acabar los problemas que se me han planteado durante el desarrollo de esta parte han sido los siguientes: el primero, sin duda, fue la complejidad de la instalación, ya que si te equivocas en alguno de los pasos sin darte cuenta las cosas no funcionan correctamente.

Es un ejemplo de esto, la instalación de Bootstrap para React, la primera opción que encontré era instalar `npm bootstrap-react` pero esta opción no funciona para el caso de utilizar webpack.

Otra dificultad, es la gestión de clicks o eventos en la página, ya que debes tener en cuenta la tipología de función o clase que estás usando para poder realizar la detección de forma correcta.

Sencillamente, el código no es ni de lejos tan simple cómo programar HTML con CSS y JS directo, sin embargo el rango de posibilidades y funciones que ofrece es muy amplio y

actualmente hay mucha gente usando React, y por lo tanto muchos componentes ya creados y de uso libre que puedes encontrar directamente en la red.

4. Resultados

Una vez finalizado el estudio de tecnologías básicas vs. React, debía escoger cuál de las dos webs creadas sería la usaríamos cómo página web oficial de OpenHuaca; había ciertas consideraciones a tener en cuenta, la primera que debía ser con código fácilmente entendible para todos los miembros del equipo y la segunda que debía incluir todas las especificaciones establecidas inicialmente a la vez que podía servir en el futuro para añadirle funcionalidades nuevas.

Barajando los pros y contras de cada una de las opciones, finalmente escogí la web de React por los siguientes motivos:

Aunque no se la web con el código más entendible en un primer momento para todo el equipo, sí que es el más útil a la hora de modificar o cambiar funciones ya que todo está separado en componentes, por lo tanto lo menos sencillo acabará siendo lo más cómodo y segundo que el hecho de que tenga un servidor instalado me permite incorporar de forma fácil y rápida mi página web al host deseado para que se muestre en la página web oficial de OpenHuaca: www.openhuaca.com.

A continuación se muestra la tabla comparativa entre las diversas tecnologías web estudiadas:

COMPARACIÓN1 : Tecnología sencilla y fácil del primer momento			
Tecnología Web	Más útil	Normal	Menos útil
JQuery			
Angular			
React			
COMPARACIÓN 2: Tecnología para sitios con múltiples páginas			
Tecnología Web	Más útil	Normal	Menos útil
JQuery			
Angular			

React			
COMPARACIÓN3 : Tecnología con código reutilizable			
Tecnología Web	Más útil	Normal	Menos útil
JQuery			
Angular			
React			
COMPARACIÓN 4: Tecnología con funcionalidades dinámicas			
Tecnología Web	Más útil	Normal	Menos útil
JQuery			
Angular			
React			
COMPARACIÓN 5: Tecnología con interacción directa con el backend			
Tecnología Web	Más útil	Normal	Menos útil
JQuery			
Angular			
React			
COMPARACIÓN 7: Tecnología en gran uso en la actualidad			
Tecnología Web	Más útil	Normal	Menos útil
JQuery			
Angular			
React			

5. Costes

Este apartado recoge el conjunto de costes que ha supuesto la realización del proyecto.

Al tratarse de un proyecto basado en el estudio y la búsqueda de información sobre tecnologías web, seguido del desarrollo de un portal web, no se ha requerido dinero para la creación de ningún prototipo ni hardware.

Por lo tanto, los gastos totales se pueden dividir en dos grupos: formación y salario. El primero engloba el conjunto de cursos realizados durante el desarrollo del proyecto y el segundo la cantidad de horas dedicadas evaluadas cómo ingeniero junior.

5.1 Formación

Udemy – The Web Developer BootCamp → 9.99€

Udemy – The Complete React Web Developer Course → 9.99€

TOTAL: 19,98€

5.2 Salario

Carla Brugulat Rica – Ingeniero Junior → 10€/hora x 8h x 5 días x 15 semanas*

(el tiempo invertido en la realización del proyecto se detalla en el apartado 4.4 Plan de Trabajo)

TOTAL: 6000€

COSTES TOTALES: 6019,98€

6. Conclusiones y futuros desarrollos

Después de todo el desarrollo del proyecto, se puede concluir que las páginas que son parte del motor que mueve nuestra sociedad y que podemos encontrar muchas formas distintas de programación para crearlas.

Dentro de todas estas opciones hay unas que son esenciales y que escojamos un lenguaje u otro siempre vamos a acabar utilizándolas y son HTML/CSS. Dentro de los lenguajes que dinamizan las webs, hoy en día, de seguro que vamos a utilizar JS, ya que PHP está poco a poco siendo menos utilizado y quedando en segundo plano.

Una vez tienes claros los conceptos básicos que además, en Internet, puedes encontrar millones de cursos que te explican a utilizar estas técnicas des de cero, yo recomiendo 100% el curso de Udemy ("Web Developer BootCamp") ya que para mí fue de gran ayuda.

Sobre React, creo que es un lenguaje de programación de futuro: colaborativo, reutilizable, dinámico pero creo que el sistema de instalación puede echar un poco para atrás sobre todo para aquellos que se inician en la programación web.

Por otra parte creo que los gestores de contenidos son una herramienta muy útil para todas aquellas empresas grandes o pequeñas que la persona que genera los contenidos del site no es programadora.

Finalmente, este proyecto no acaba aquí, la experiencia continúa con el equipo de OpenHuaca con el que pretendo seguir colaborando para crear finalmente el site que ellos necesiten a medida que el software vaya avanzando; los siguientes pasos a realizar que ya están pensados son la creación de un usuario de contacto y una zona de administración en la web, para todo aquel que forme parte del equipo pueda modificar sus máquinas y contenedores a través de la web.

Bibliografia

- [1] "W3Schools.com". [Online] Available <https://www.w3schools.com/>. [Accessed: 10 Maig 2018].
- [2] "MDN web docs". [Online] Available: <https://developer.mozilla.org/es/> [Accessed: 10 Maig 2018].
- [3] "Pixabay". [Online] Available: <https://pixabay.com/> [Accessed: 10 Maig 2018].
- [4] "w3Techs". [Online] Available: <https://w3techs.com/> [Accessed: 10 Maig 2018].
- [5] "Oracle Help Center". [Online] Available: https://docs.oracle.com/cd/E29542_01/doc.11111/e35835/wemrestresources.htm#WBCSR354 [Accessed: 10 Maig 2018].
- [6] "Google Fonts". [Online] Available: <https://fonts.google.com/> [Accessed: 10 Maig 2018].
- [7] "Colours NeilOrangePeel [Online] Available: <http://colours.neilorangepeel.com/> [Accessed: 10 Maig 2018].
- [8] "Udemy-Web Developer BootCamp". [Online] Available: <https://www.udemy.com/the-web-developer-bootcamp/> [Accessed: 10 Maig 2018].
- [9] "Udemy – React 2nd Edition". [Online] Available: <https://www.udemy.com/react-2nd-edition/> [Accessed: 10 Maig 2018].
- [10] "Babel". [Online] Available: <http://babeljs.io/> [Accessed: 10 Maig 2018].
- [11] "Node.js". [Online] Available: <https://nodejs.org/en/> [Accessed: 10 Maig 2018].
- [12] "npm.js". [Online] Available: <https://www.npmjs.com/> [Accessed: 10 Maig 2018].
- [13] "Yarn Package". [Online] Available: <https://yarnpkg.com/lang/en/> [Accessed: 10 Maig 2018].

Glosario

Listado de los acrónimos junto su significado que aparecen en la tesis:

- (1) HTML → HyperText Markup Language (Lenguaje de marcas de hipertexto)
- (2) CSS → Cascading Stylesheets (Hojas de estilo en cascada)
- (3) JS → JavaScript
- (4) OWCS → Oracle WebCenter Sites
- (5) PHP → PHP Hypertext Preprocessor (procesador de hipertexto)
- (6) ASP → Active Server Pages
- (7) JSP → JavaServer Pages
- (8) JSTL → Java Server Page Standard Tag Library